MCP：AIアプリをシンプルに

MODEL CONTEXT PROTOCOL（MCP）入門



James Chapman
AI Curriculum Manager, DataCamp

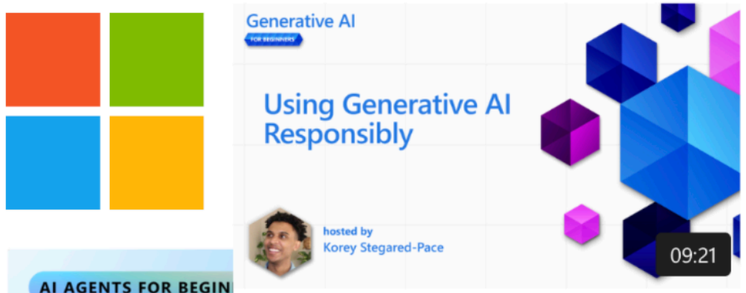
datacamp

講師紹介



Korey Stegared-Pace

Senior AI Cloud Advocate at Microsoft





EPISODE

Using Generative AI Responsibly (Part 3 of 18) | Generative AI for Beginners

Jun 25, 2024

Azure

EPISODE

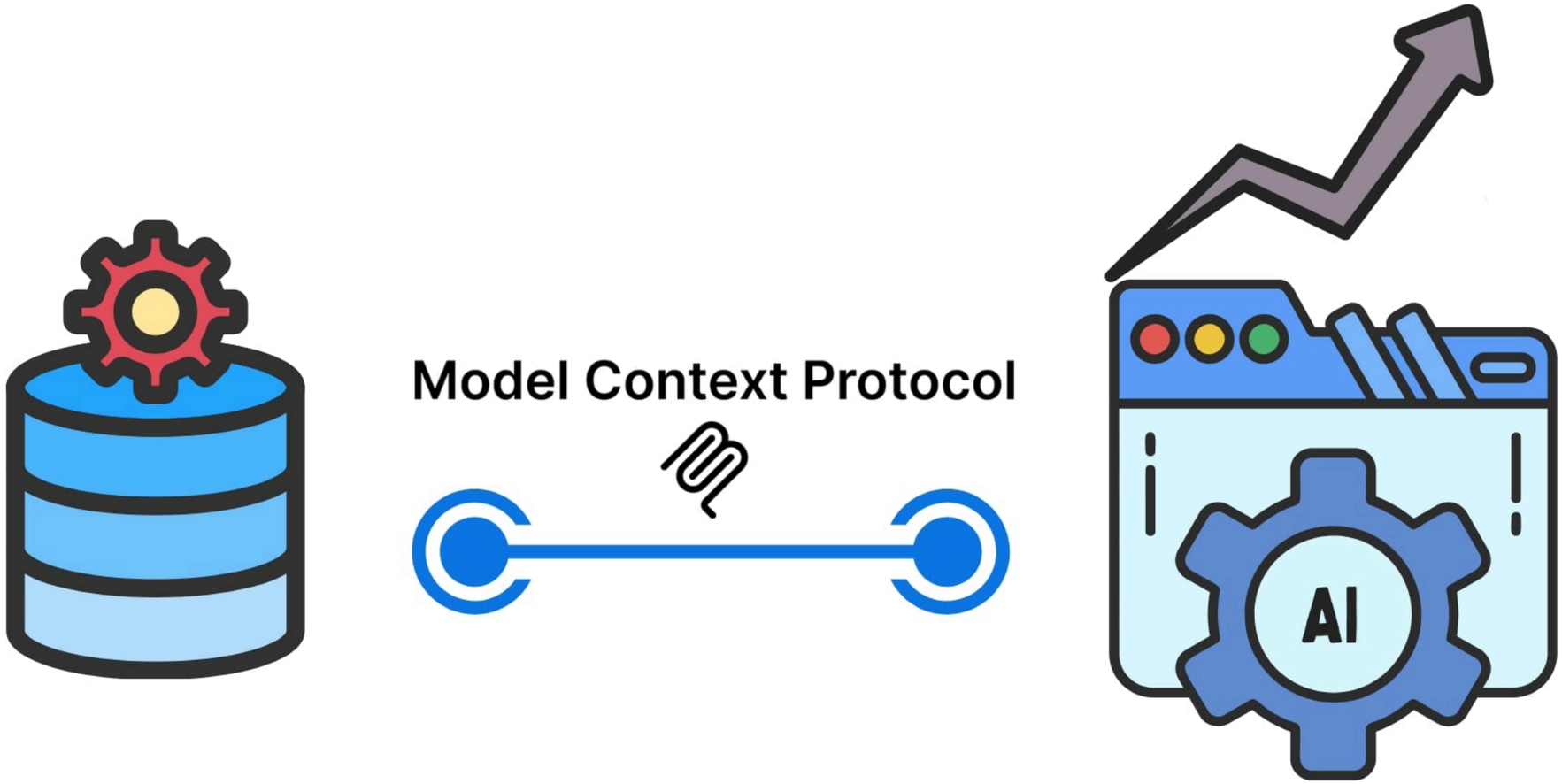
What are AI agents? (Part 1 of 10) | AI Agents for Beginners

Mar 24, 2025

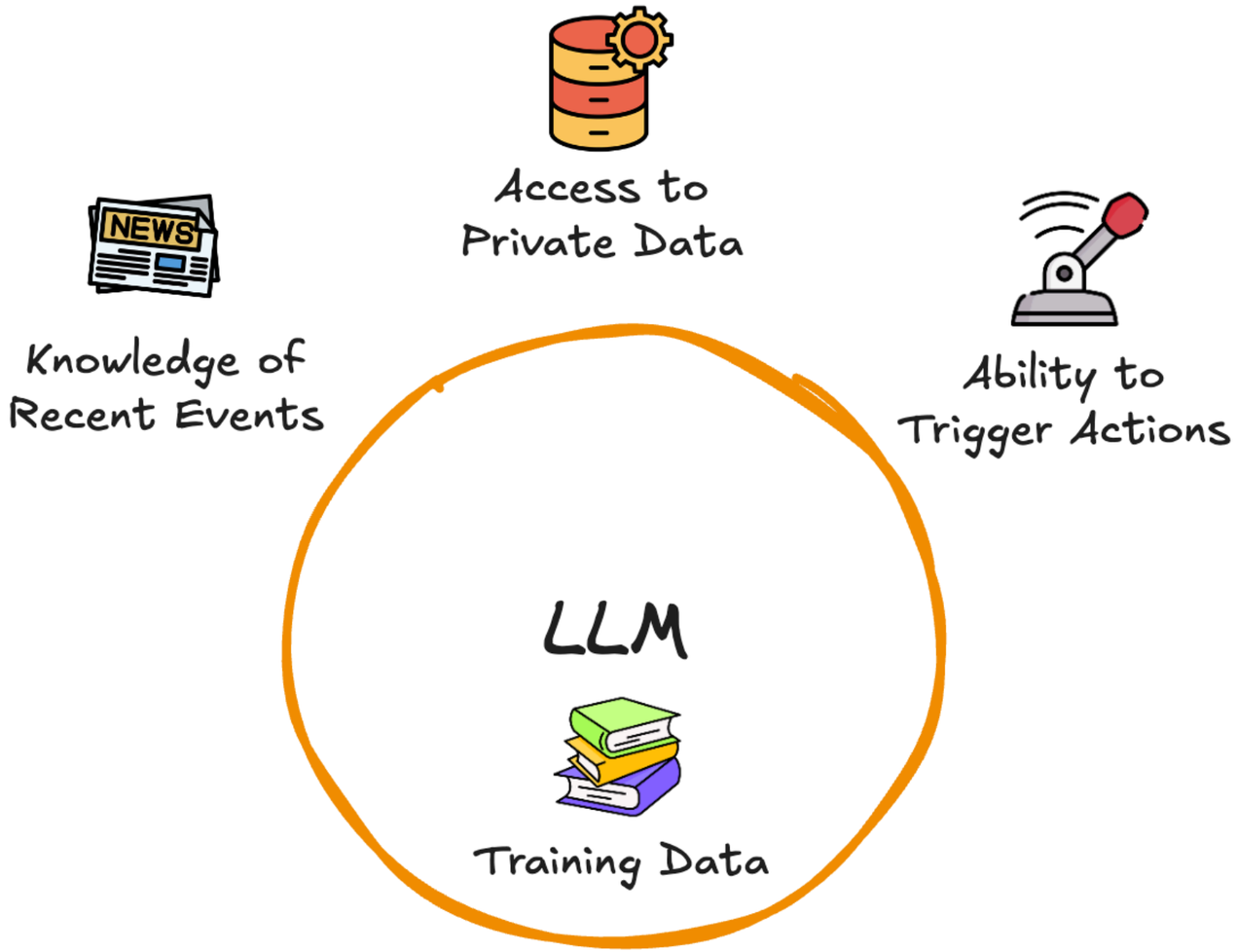
Beginner

[1] https://developer.microsoft.com/en-us/advocates/korey-stegared-pace

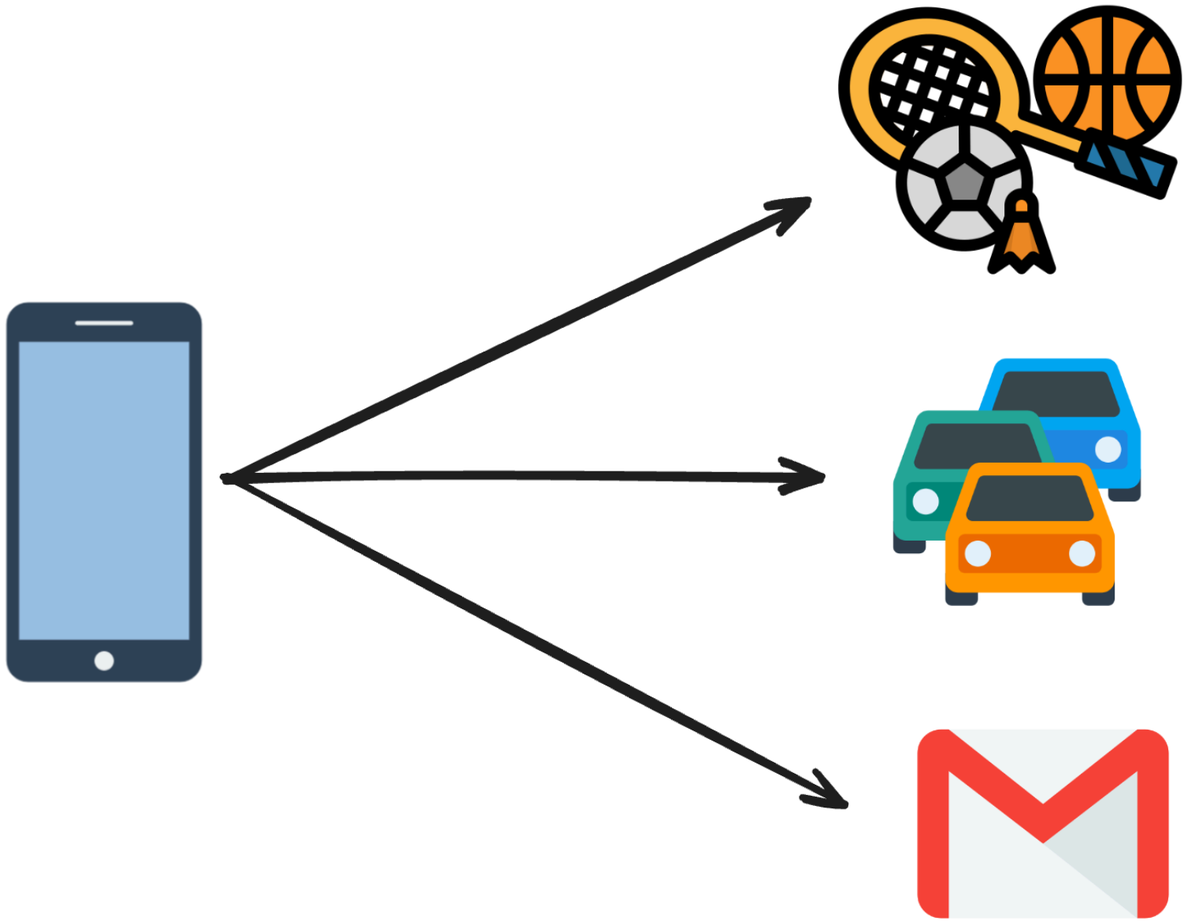
コースの概要



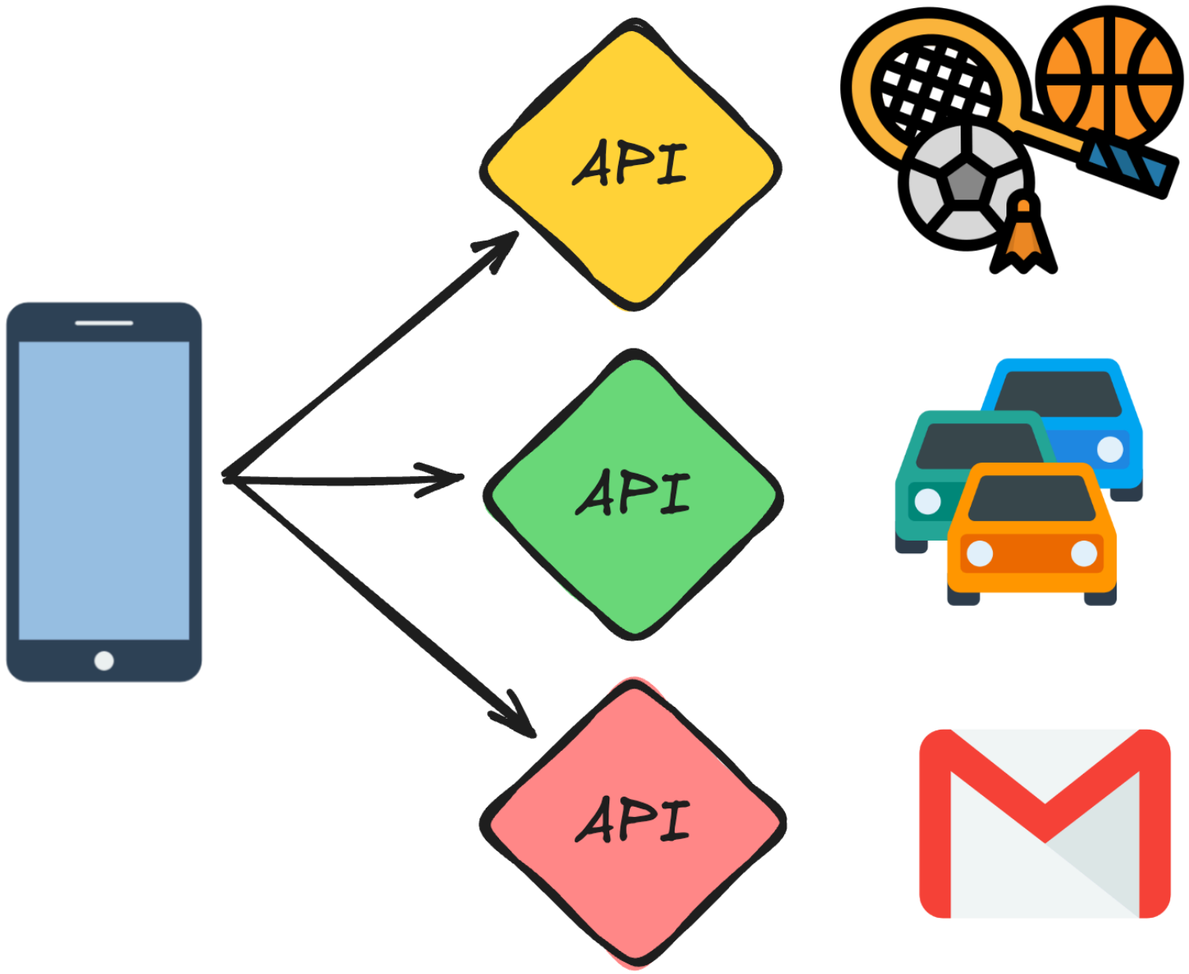
LLMの大きな課題


Access to
Private Data
NEWS
Knowledge of
Recent Events
Ability to
Trigger Actions
LLM
Training Data

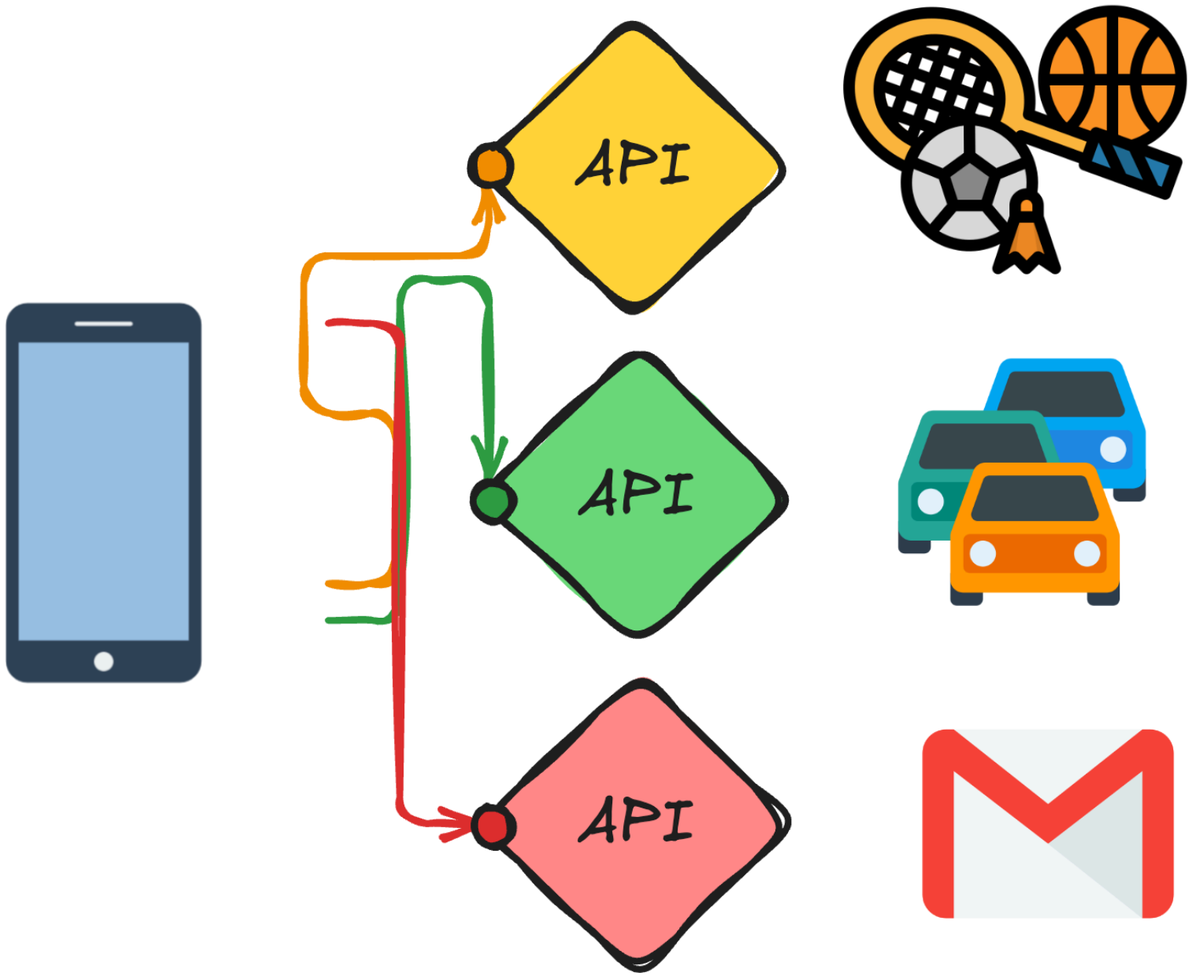
LLMの大きな課題



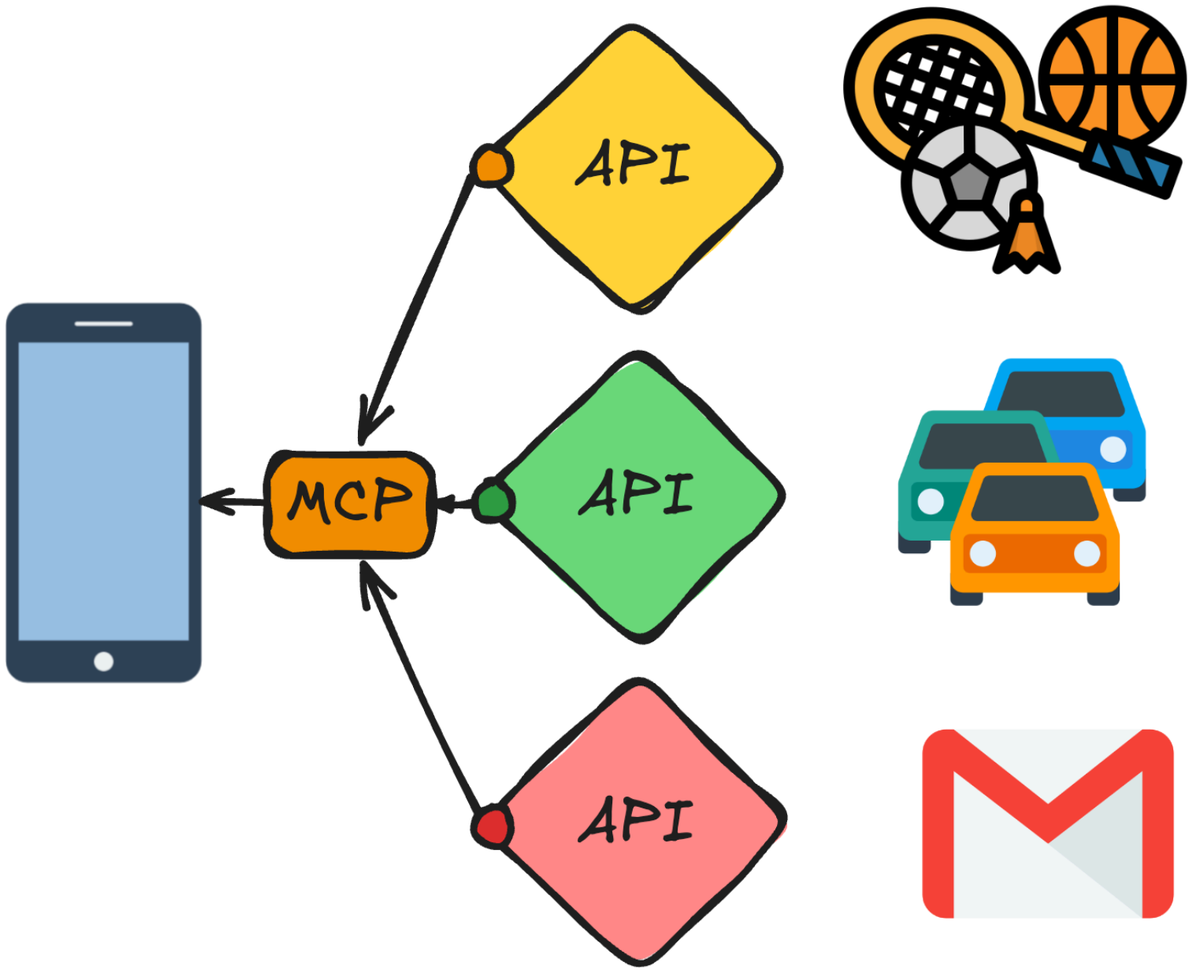
LLMの大きな課題



LLMの大きな課題



MCPが解決策に！

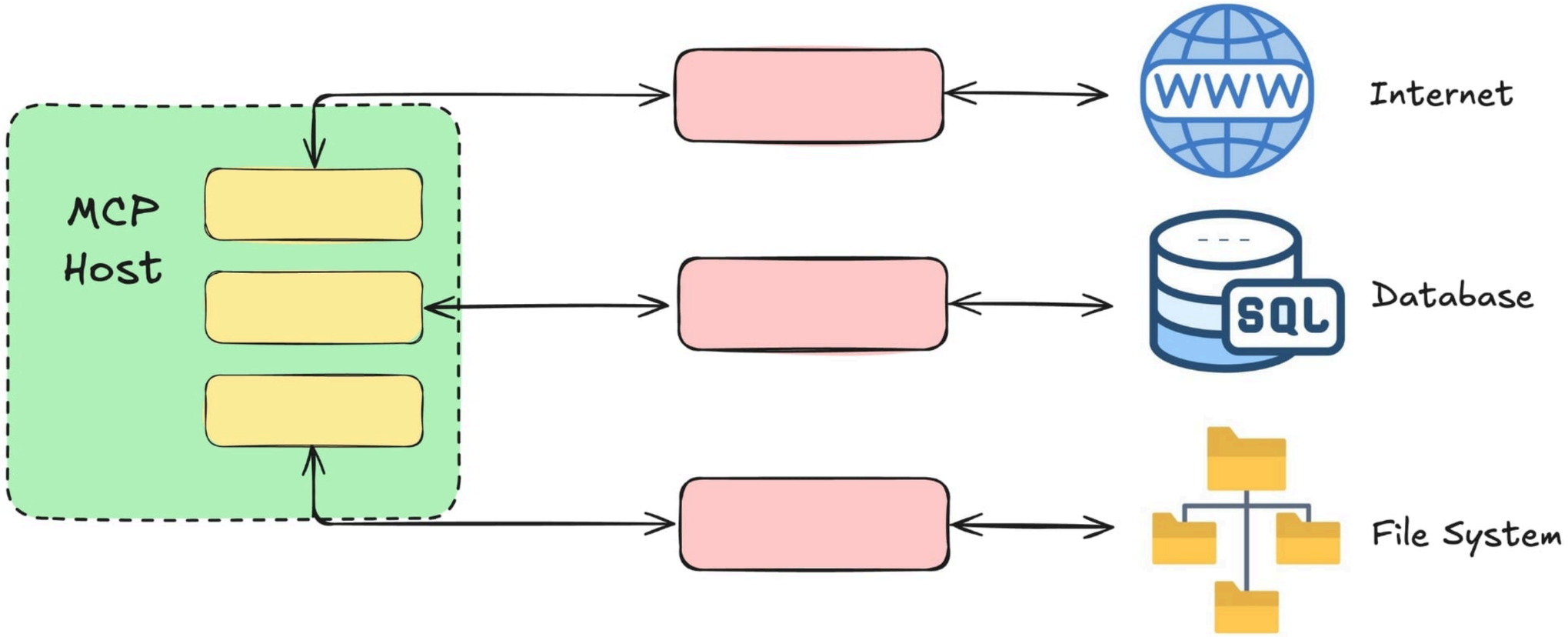


MCPが解決策に！

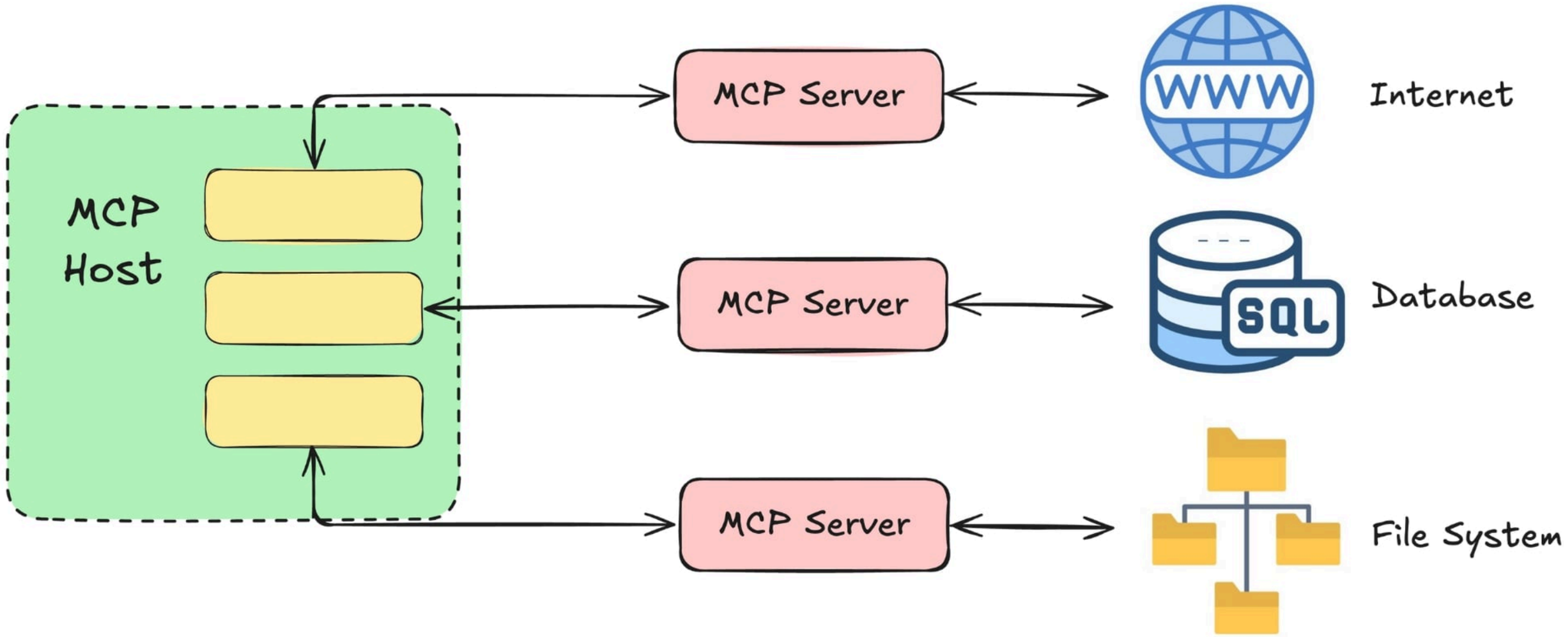
- **接続性を簡素化**し、機能を追加
  - *動的ツール検出* → AIが利用可能な機能を「認識」
- MCPのコンポーネント：
  - ホスト、クライアント、サーバー



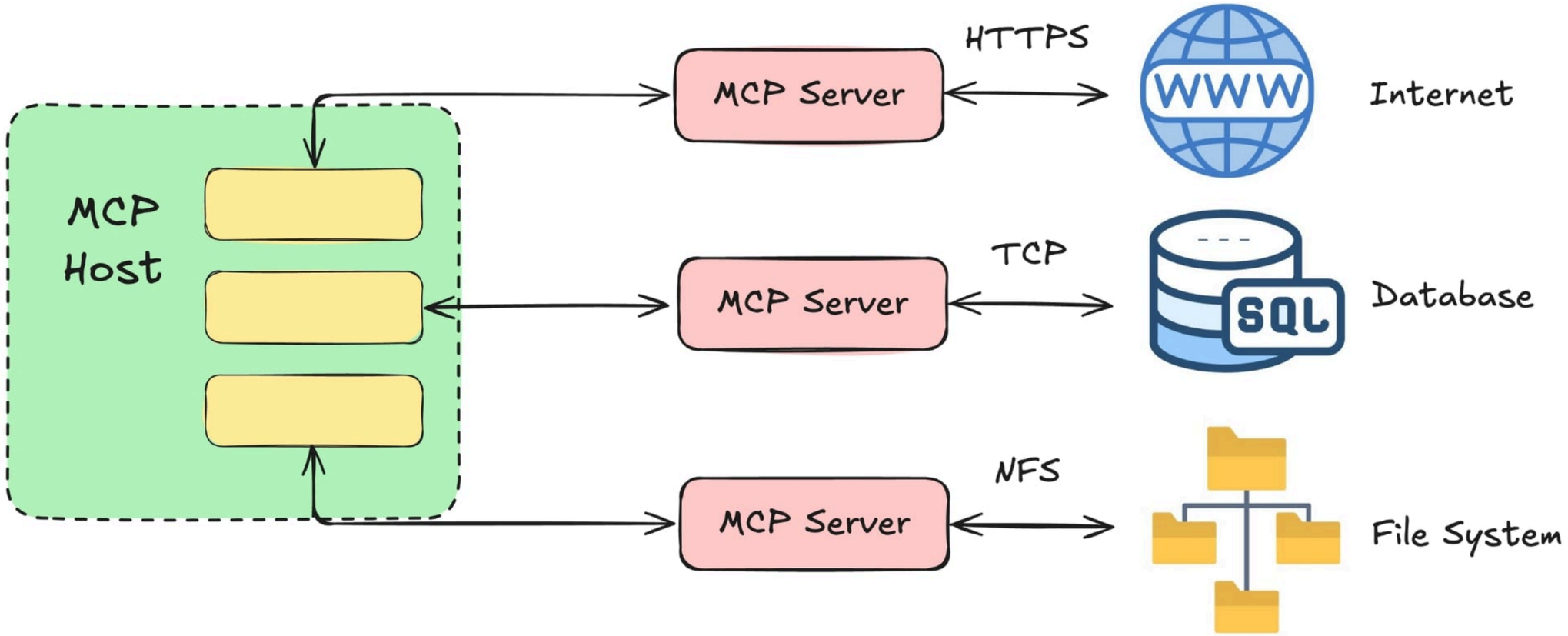
MCPのアーキテクチャ



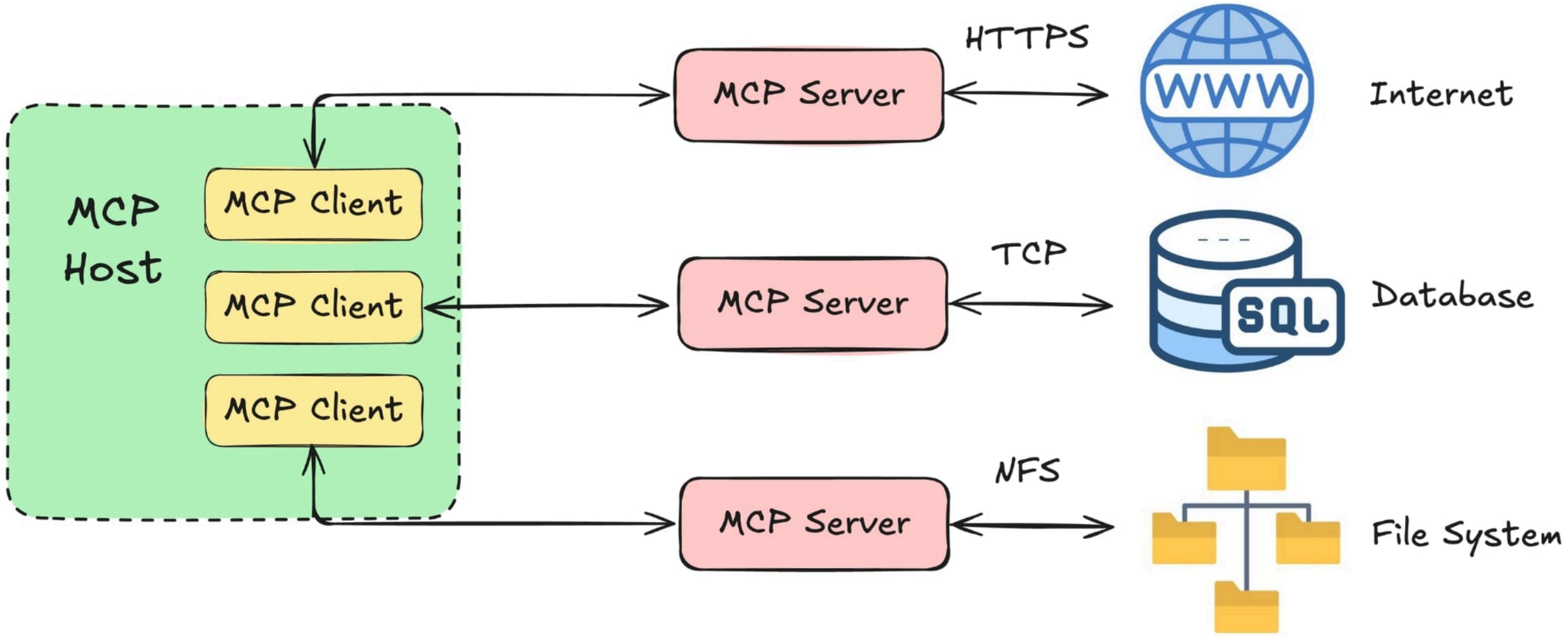
MCPのアーキテクチャ



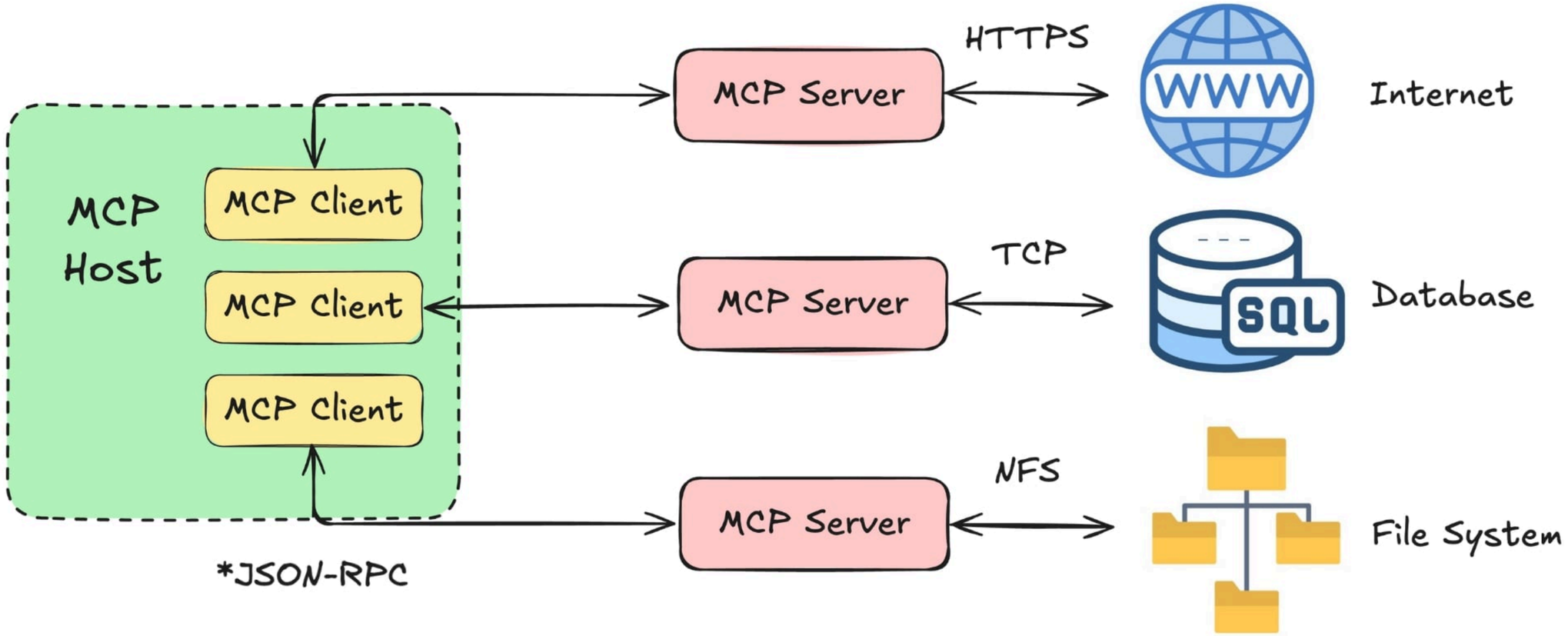
MCPのアーキテクチャ



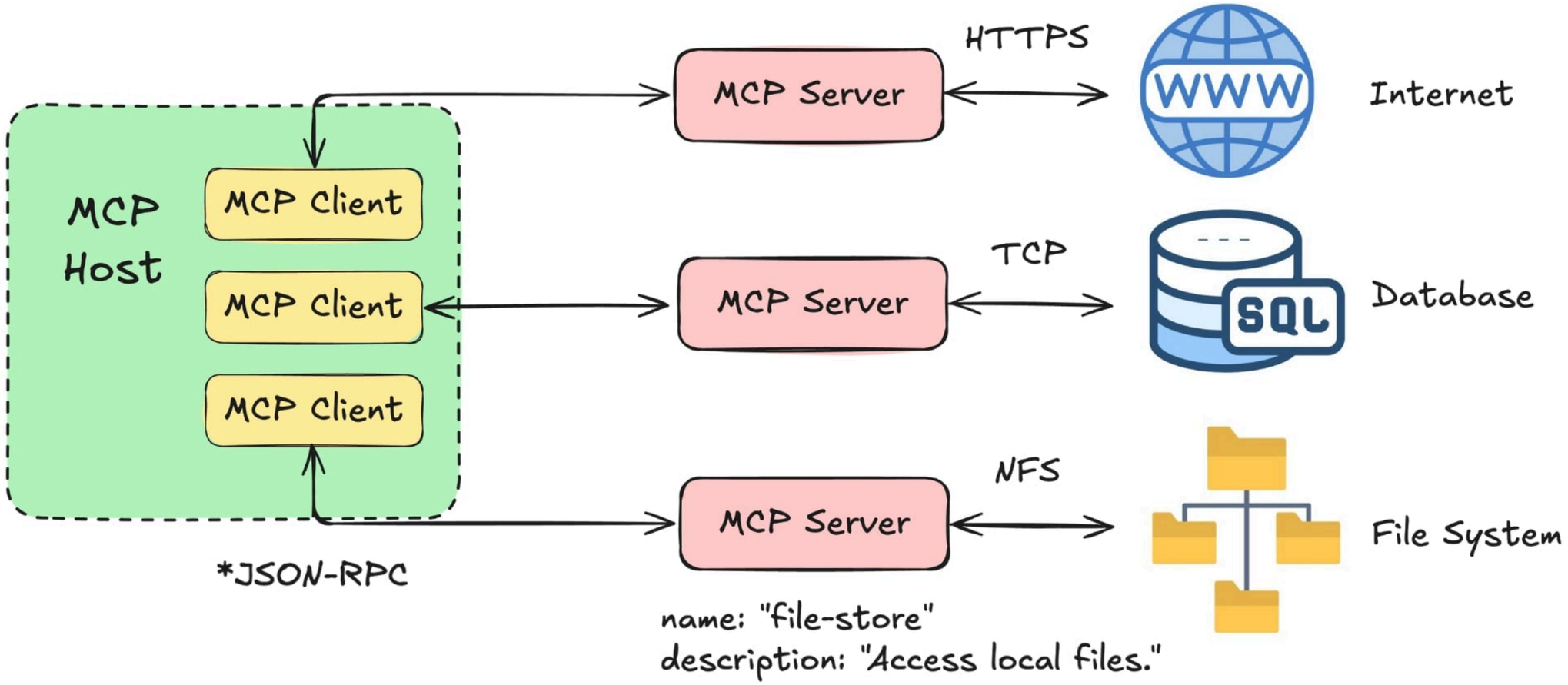
MCPのアーキテクチャ



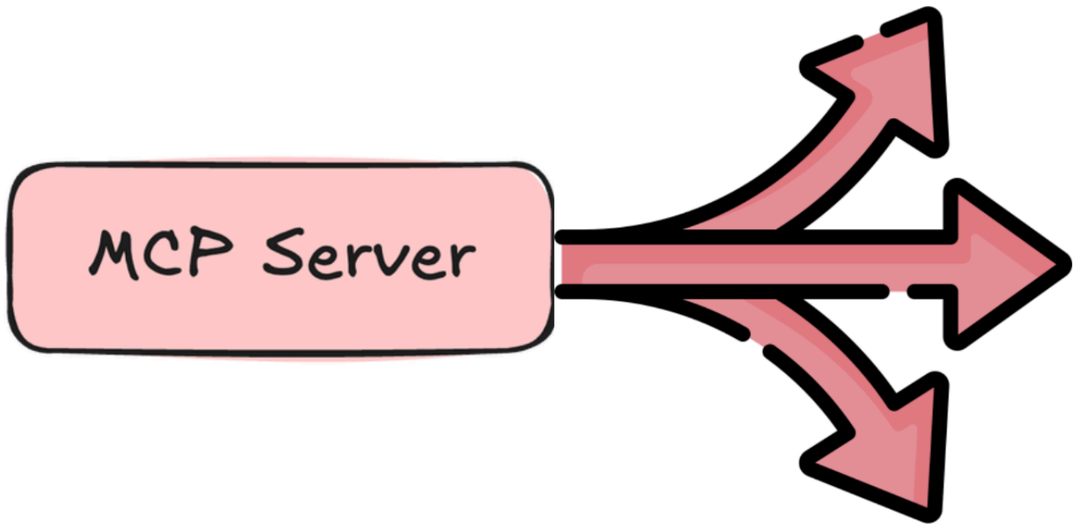
MCPのアーキテクチャ



MCPのアーキテクチャ



プリミティブ



Tools

-> Actions the LLM can perform
-> Querying a database, booking a flight, running code

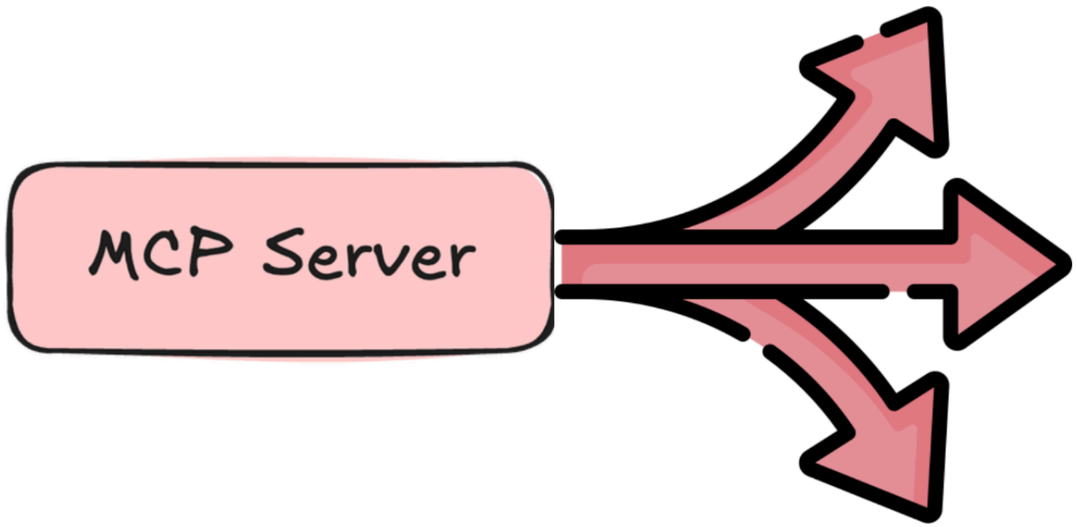
Resources

-> Read-only data or context
-> Documents, files, database records

Prompts

-> Pre-defined workflows and instructions
-> Saves users from needing to write long complex prompts

プリミティブ



Tools

-> Actions the LLM can perform
-> Querying a database, booking a flight, running code

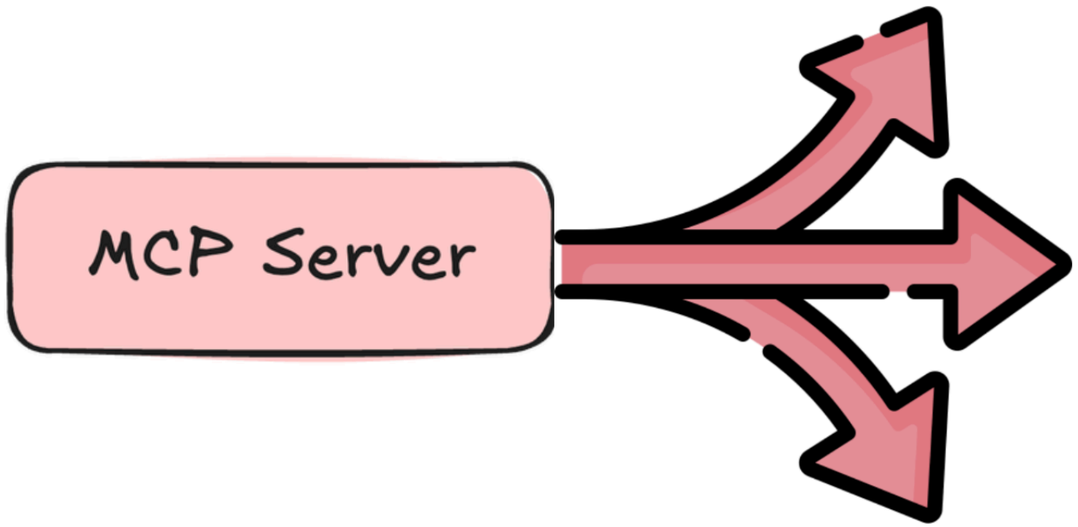
Resources

-> Read-only data or context
-> Documents, files, database records

Prompts

-> Pre-defined workflows and instructions
-> Saves users from needing to write long complex prompts

プリミティブ



Tools

-> Actions the LLM can perform
-> Querying a database, booking a flight, running code

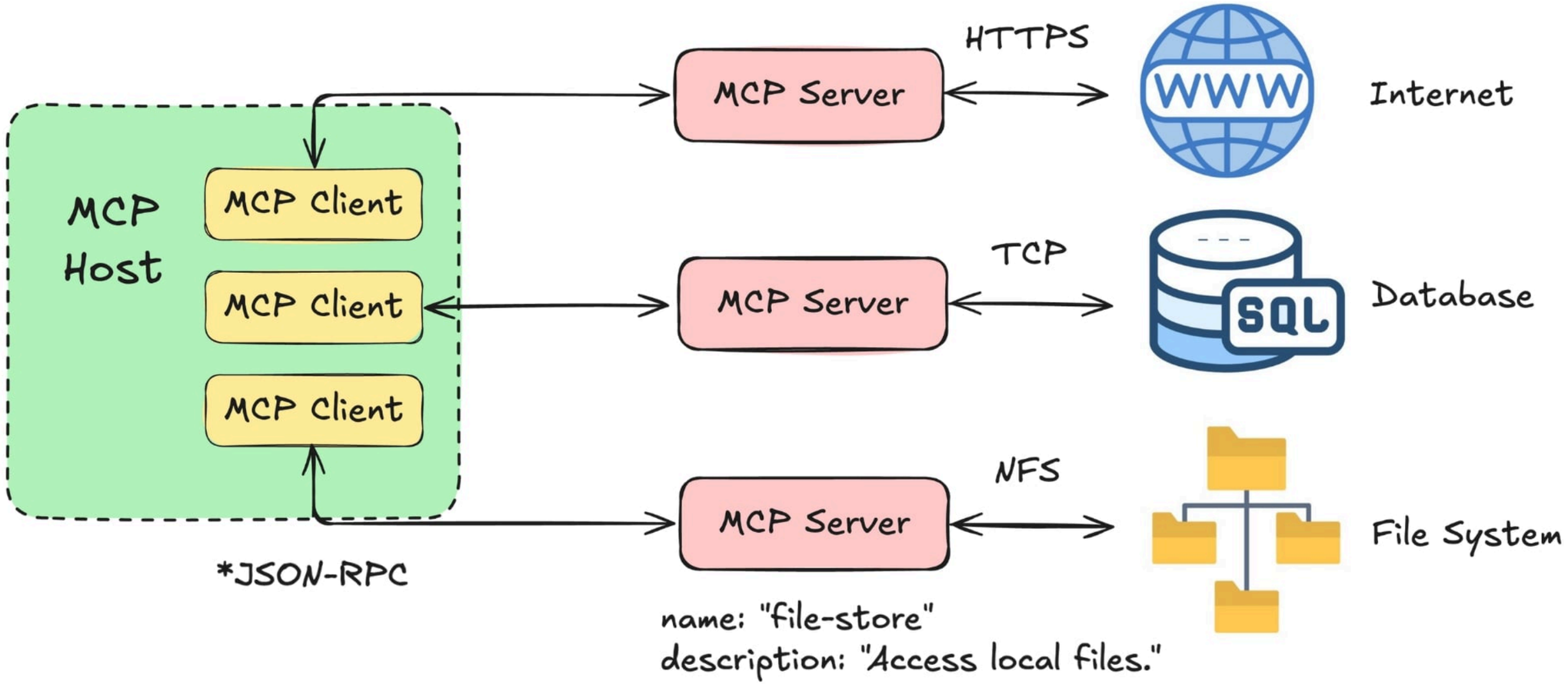
Resources

-> Read-only data or context
-> Documents, files, database records

Prompts

-> Pre-defined workflows and instructions
-> Saves users from needing to write long complex prompts

MCPのアーキテクチャ



練習しましょう！

MODEL CONTEXT PROTOCOL（MCP）入門

初めてのMCPサーバー構築

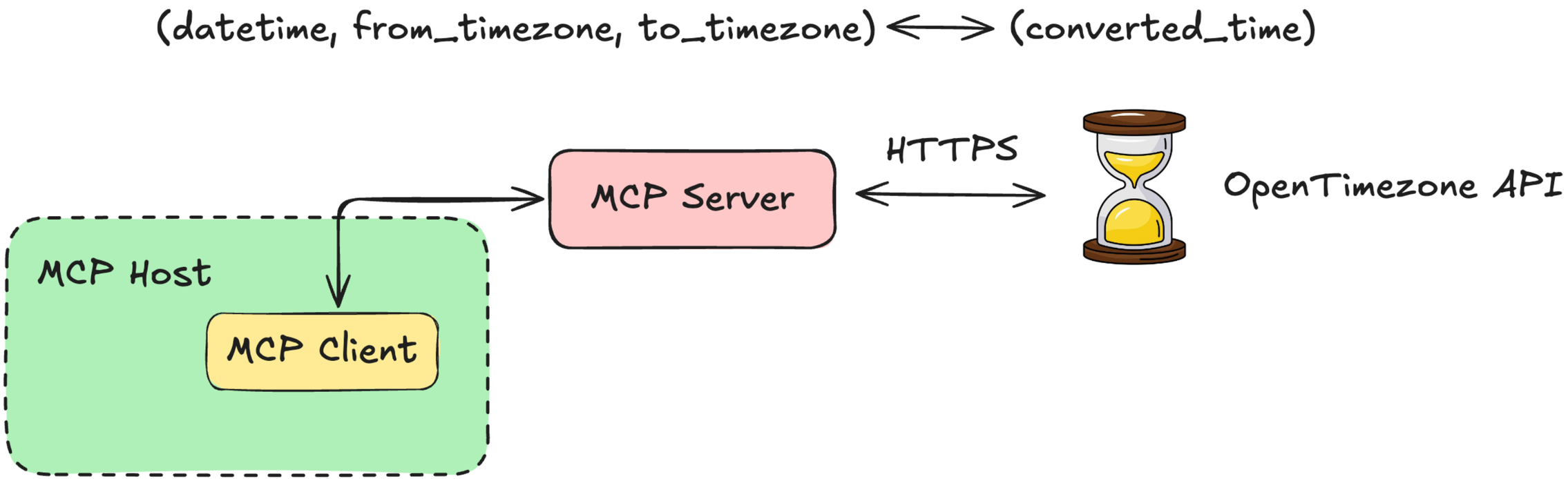
MODEL CONTEXT PROTOCOL（MCP）入門



James Chapman
AI Curriculum Manager, DataCamp

datacamp

タイムゾーンの変換



[1] https://opentimezone.com/

```python
# Create an MCP server instance
mcp = FastMCP("Timezone and Currency Converter")

@mcp.tool()
def convert_timezone(date_time: str, from_timezone: str, to_timezone: str) -> str:
    """
    Convert a datetime from one timezone to another.

    Args:
        date_time: The datetime string in ISO format (e.g., '2025-01-20T14:30:00')
        from_timezone: Source timezone (e.g., 'America/New_York')
        to_timezone: Target timezone (e.g., 'Europe/London')

    Returns:
        A string with the converted datetime and timezone information
    """
    # Define the API endpoint and format input data
    url = "https://api.opentimezone.com/convert"

    payload = {
        "dateTime": date_time,
        "fromTimezone": from_timezone,
        "toTimezone": to_timezone
    }

    # Make the API request
    response = requests.post(url, json=payload)

    data = response.json()
    converted_time = data.get('dateTime', 'N/A')

    return f"Time in {to_timezone}: {converted_time}"
```

Instantiate MCP Server

Define Tools

Doctrings and Typing

ステップ1：サーバーのインスタンス化

```python
from mcp.server.fastmcp import FastMCP
import requests

# Create an MCP server instance
mcp = FastMCP("Timezone Converter")
```

[1] https://github.com/modelcontextprotocol/python-sdk

ステップ2：ツールの追加

```
from mcp.server.fastmcp import FastMCP
import requests

# Create an MCP server instance
mcp = FastMCP("Timezone Converter")

@mcp.tool()
def convert_timezone(date_time, from_timezone, to_timezone):
```

ステップ2：ツールの追加

```python
@mcp.tool()
def convert_timezone(date_time, from_timezone, to_timezone):

    # Define the API endpoint and format input data
    url = "https://api.opentimezone.com/convert"
    payload = {"dateTime": date_time, "fromTimezone": from_timezone, "toTimezone": to_timezone}

    response = requests.post(url, json=payload)
    data = response.json()
    converted_time = data.get('dateTime', 'N/A')
    return f"Time in {to_timezone}: {converted_time}"
```

ステップ3：Docstringと型ヒント

```python
@mcp.tool()
def convert_timezone(date_time: str, from_timezone: str, to_timezone: str) -> str:
    """
    Convert a datetime from one timezone to another.

    Args:
        date_time: The datetime string in ISO format (e.g., '2025-01-20T14:30:00')
        from_timezone: Source timezone (e.g., 'America/New_York')
        to_timezone: Target timezone (e.g., 'Europe/London')

    Returns:
        A string with the converted datetime and timezone information
    """
```

```python
# Create an MCP server instance
mcp = FastMCP("Timezone and Currency Converter")

@mcp.tool()
def convert_timezone(date_time: str, from_timezone: str, to_timezone: str) -> str:
    """
    Convert a datetime from one timezone to another.

    Args:
        date_time: The datetime string in ISO format (e.g., '2025-01-20T14:30:00')
        from_timezone: Source timezone (e.g., 'America/New_York')
        to_timezone: Target timezone (e.g., 'Europe/London')

    Returns:
        A string with the converted datetime and timezone information
    """
    # Define the API endpoint and format input data
    url = "https://api.opentimezone.com/convert"

    payload = {
        "dateTime": date_time,
        "fromTimezone": from_timezone,
        "toTimezone": to_timezone
    }

    # Make the API request
    response = requests.post(url, json=payload)

    data = response.json()
    converted_time = data.get('dateTime', 'N/A')

    return f"Time in {to_timezone}: {converted_time}"
```

Instantiate MCP Server

Define Tools

Doctrings and Typing

では、練習しましょう！

MODEL CONTEXT PROTOCOL（MCP）入門

MCPクライアント・サーバー通信

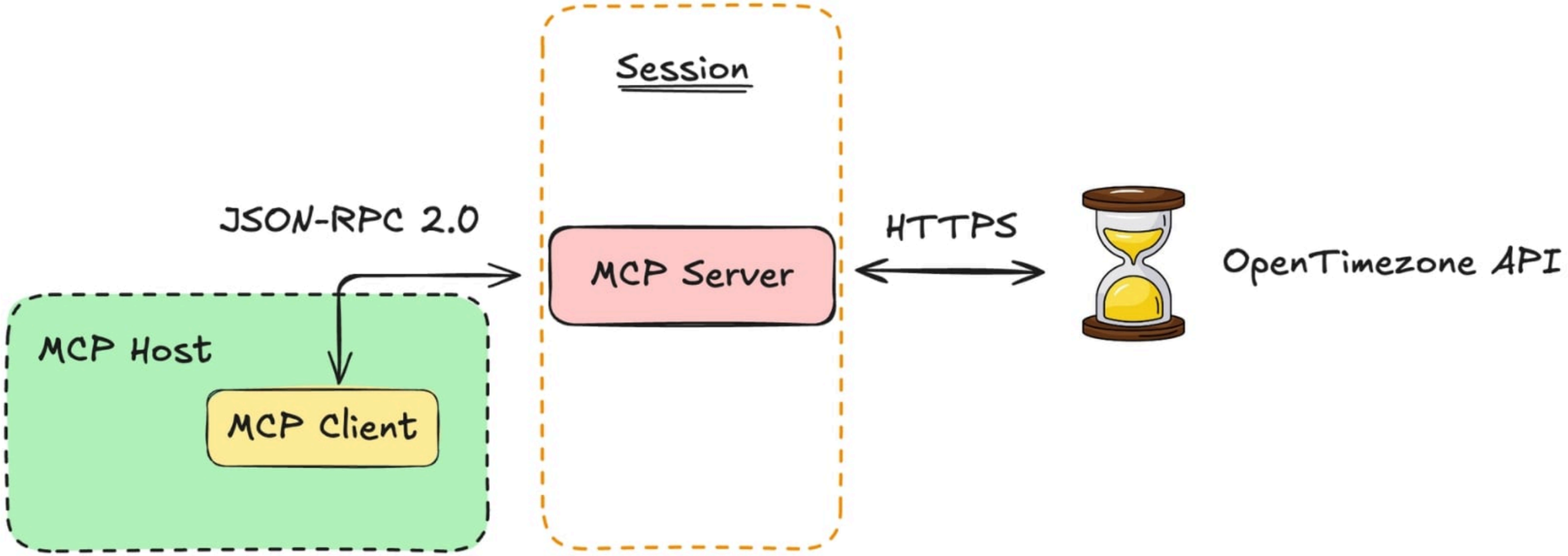
MODEL CONTEXT PROTOCOL（MCP）入門



James Chapman
AI Curriculum Manager, DataCamp

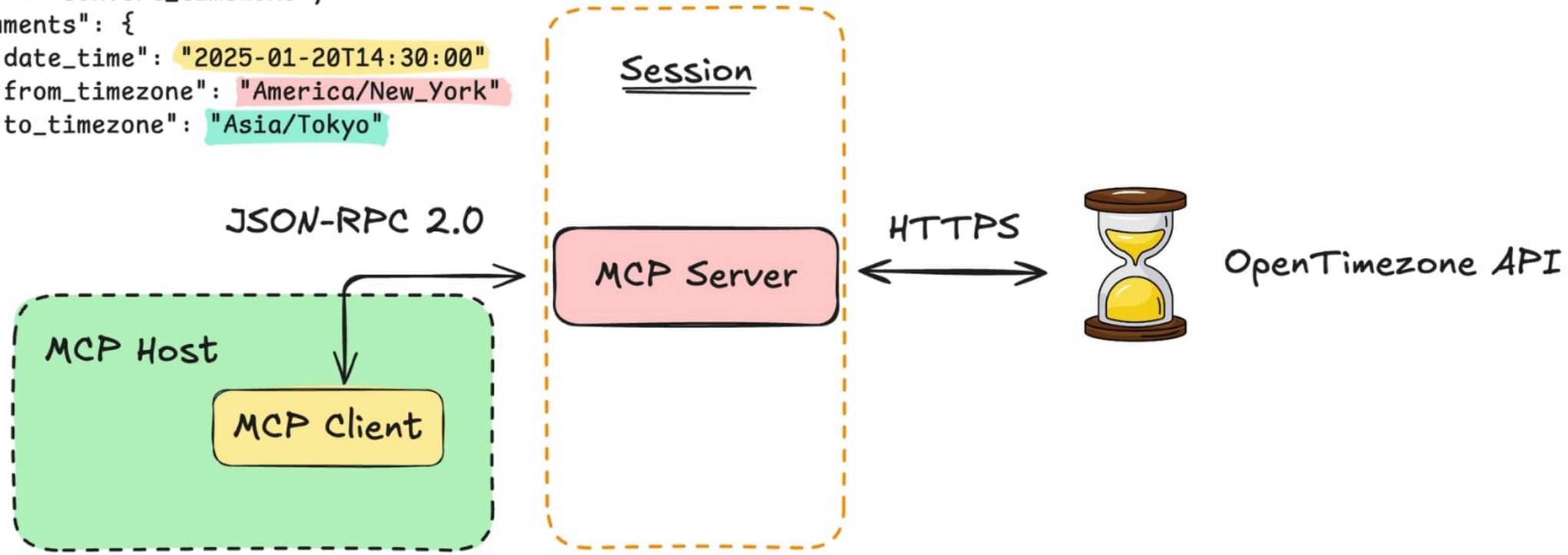
datacamp

クライアント・サーバー通信

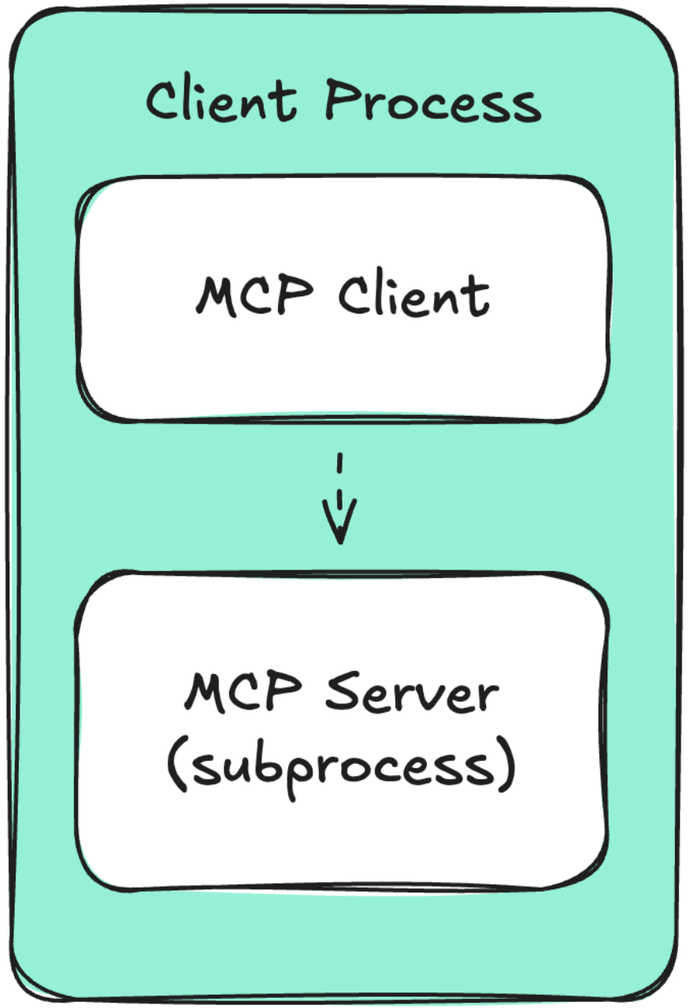


クライアント・サーバー通信

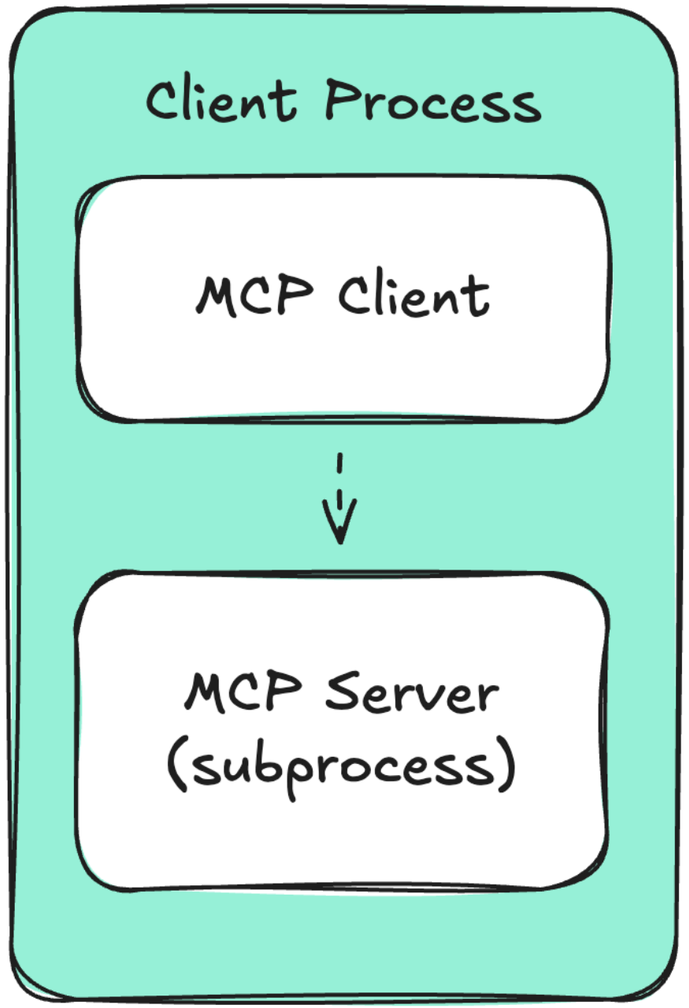
```
{
    "jsonrpc": "2.0",
    "id": 1,
    "method": "tools/call",
    "params": {
        "name": "convert_timezone",
        "arguments": {
            "date_time": "2025-01-20T14:30:00"
            "from_timezone": "America/New_York"
            "to_timezone": "Asia/Tokyo"
        }
    }
}
```



トランスポート：標準入出力（stdio）

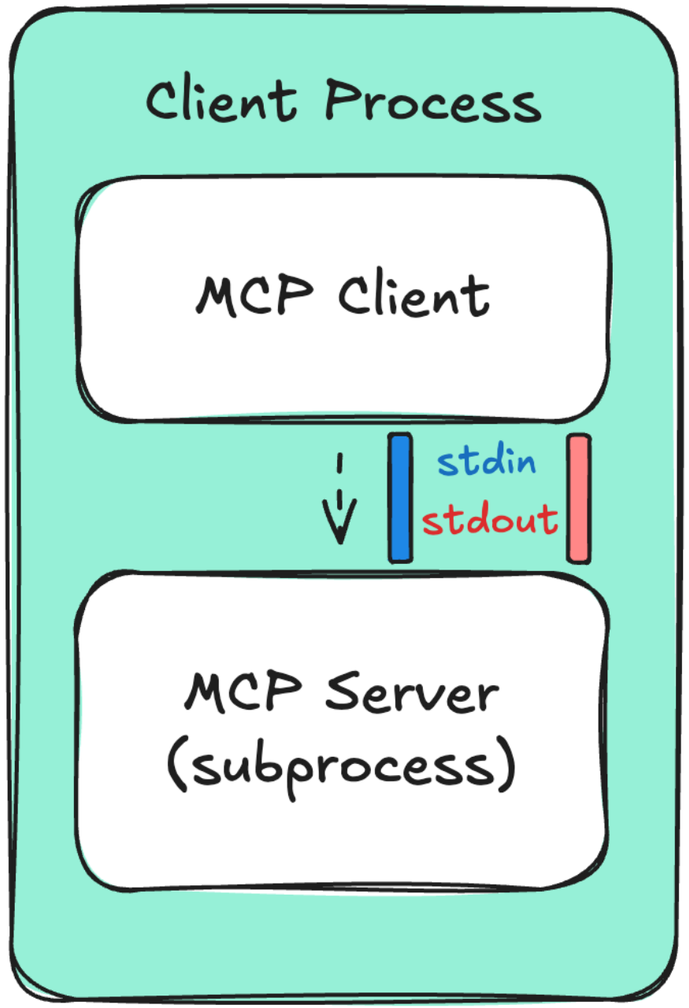


トランスポート：標準入出力（stdio）



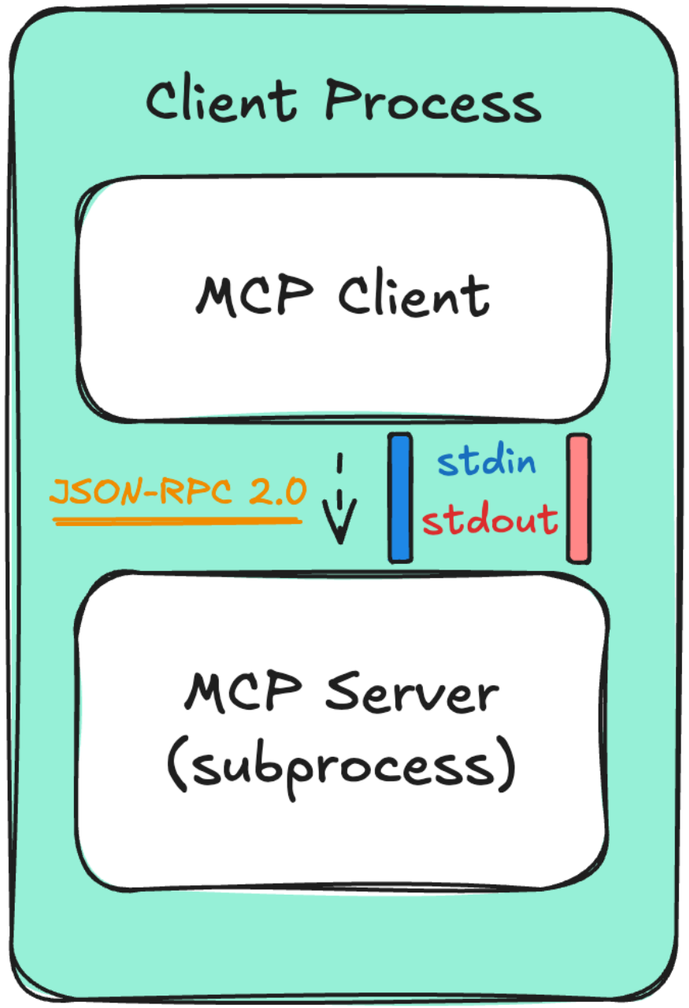
- サーバーは**子プロセス**として実行

トランスポート：標準入出力（stdio）



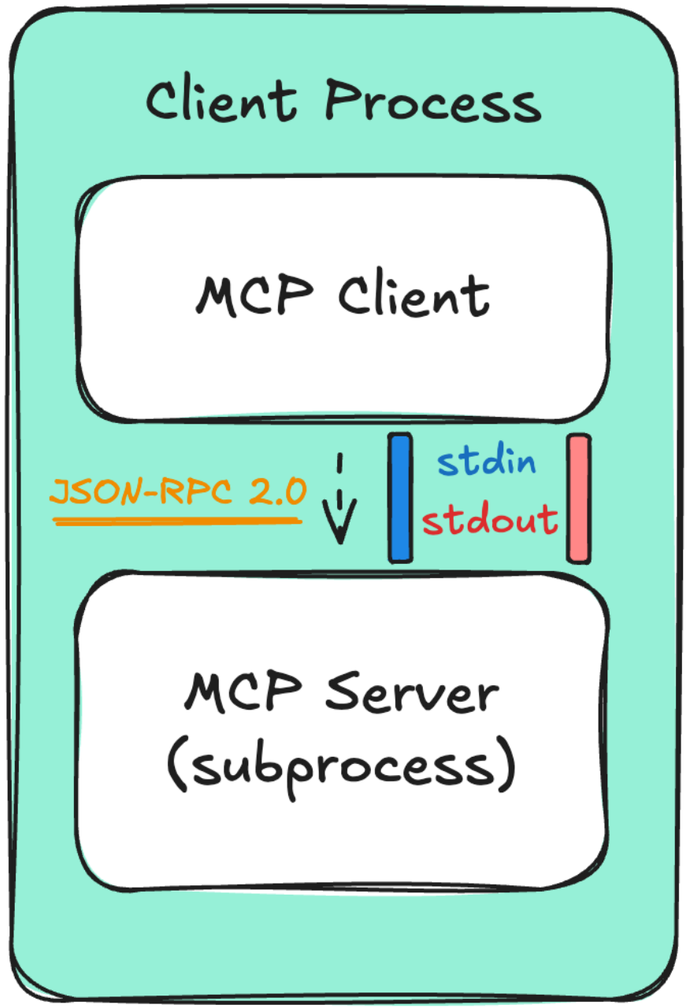
- サーバーは**子プロセス**として実行
- パイプ（**stdin/stdout**）で通信

トランスポート：標準入出力（stdio）



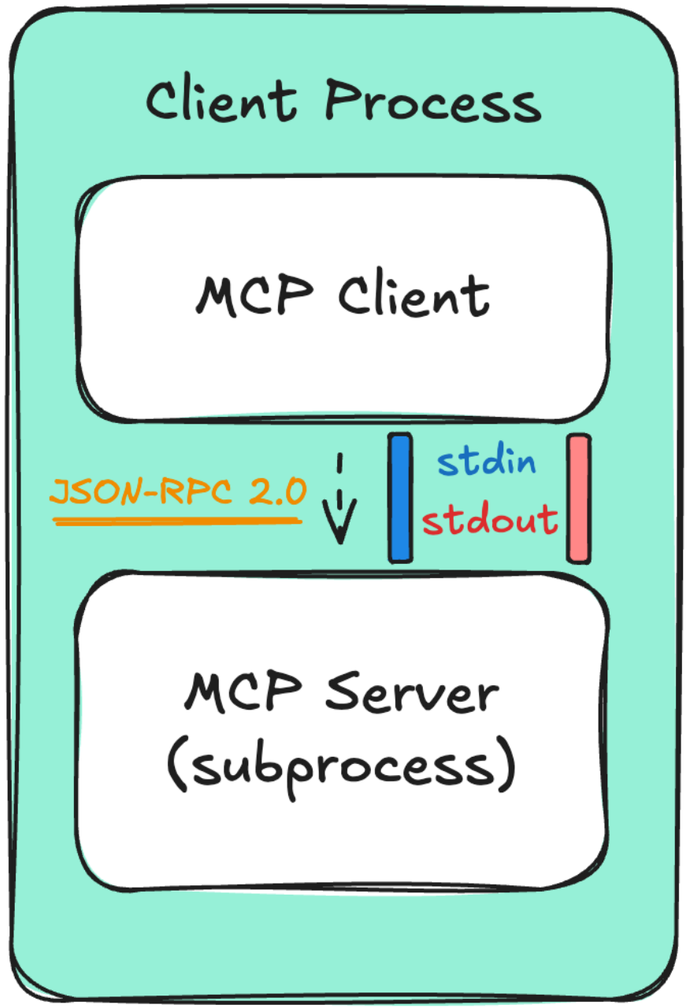
- サーバーは**子プロセス**として実行
- パイプ（**stdin/stdout**）で通信

トランスポート：標準入出力（stdio）



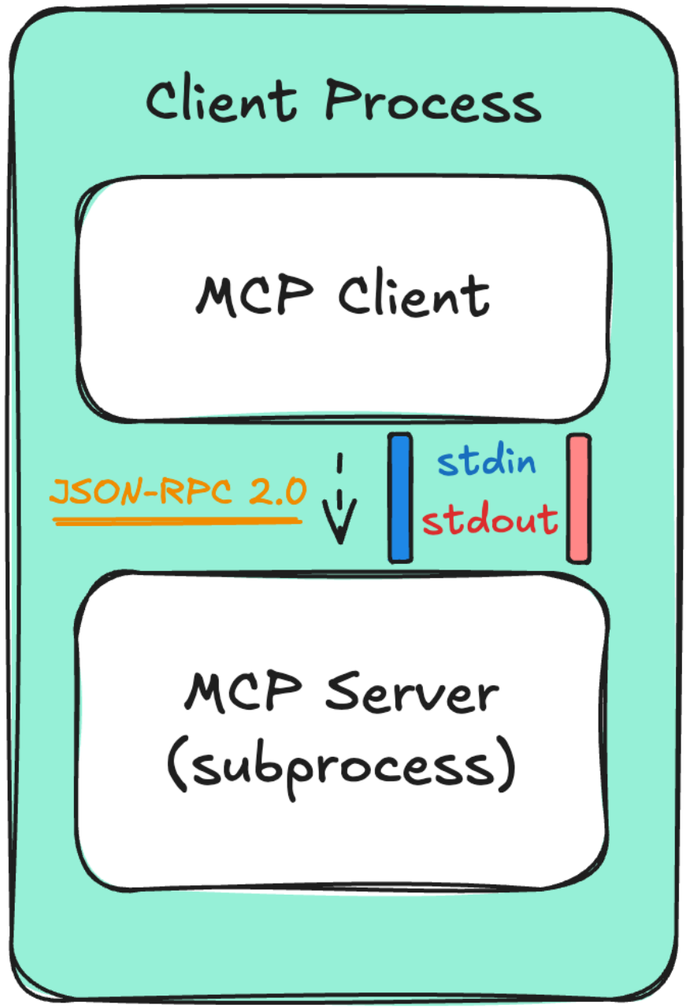
- サーバーは**子プロセス**として実行
- パイプ（**stdin/stdout**）で通信
- *ローカルのみ* - 同一マシン

トランスポート：標準入出力（stdio）



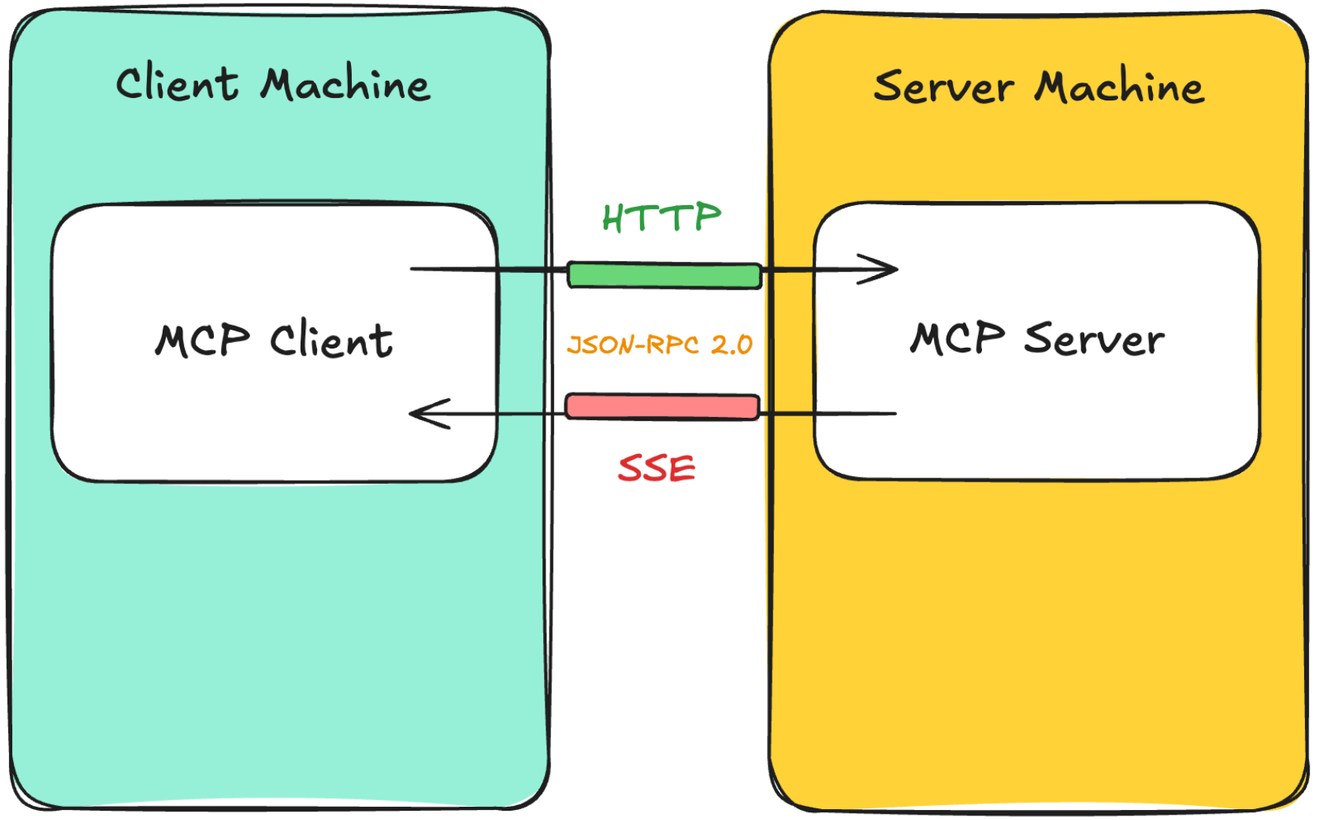
- サーバーは**子プロセス**として実行
- パイプ（**stdin/stdout**）で通信
- *ローカルのみ* - 同一マシン
- シンプル、ネットワーク設定不要

トランスポート：標準入出力（stdio）



- サーバーは**子プロセス**として実行
- パイプ（**stdin/stdout**）で通信
- *ローカルのみ* - 同一マシン
- シンプル、ネットワーク設定不要
- **クライアント終了時にサーバーも停止**

トランスポート：Streamable HTTP



MCPサーバー：timezone_server.py

```python
import mcp
from mcp.server.fastmcp import FastMCP
import requests

mcp = FastMCP("Timezone Converter")

@mcp.tool()
def convert_timezone(date_time: str, from_timezone: str, to_timezone: str) -> str:
    ...

if __name__ == "__main__":
    mcp.run(transport="stdio")
```

MCPクライアント：サーバーツールの一覧取得

```python
from mcp import ClientSession, StdioServerParameters
from mcp.client.stdio import stdio_client

async def get_tools_from_mcp():
    # Define the server parameters
    params = StdioServerParameters(command=sys.executable, args=["timezone_server.py"])

    # Connect to the MCP server and open a session
    async with stdio_client(params) as (reader, writer):
        async with ClientSession(reader, writer) as session:
            # Initialize the session
            await session.initialize()
            # Ask the server what tools it provides
            response = await session.list_tools()
```

MCPクライアント：サーバーツールの一覧取得

```
async def get_tools_from_mcp():
            # ...

            # Ask the server what tools it provides
            response = await session.list_tools()
            print("Connected to MCP server!")
            print("Available tools:")
            for tool in response.tools:
                print(f" - {tool.name}: {tool.description}")

            return response.tools

asyncio.run(get_tools_from_mcp())
```

MCPクライアント：サーバーツールの一覧取得

```
Connected to MCP server!
Available tools:
 - convert_timezone:
    Convert a datetime from one timezone to another.

    Args:
        date_time: The datetime string in ISO format (e.g., '2025-01-20T14:30:00')
        from_timezone: Source timezone (e.g., 'America/New_York')
        to_timezone: Target timezone (e.g., 'Europe/London')

    Returns:
        A string with the converted datetime and timezone information
```

MCPクライアント：サーバーツールの呼び出し

```python
async def call_mcp_tool(tool_name: str, arguments: dict) -> str:
    params = StdioServerParameters(command=sys.executable, args=["timezone_server.py"])

    async with stdio_client(params) as (reader, writer):
        async with ClientSession(reader, writer) as session:

            await session.initialize()

            result = await session.call_tool(tool_name, arguments)

            text_content = result.content[0].text

            print(f"Conversion Result: {text_content}")
            return str(text_content)
```

MCPクライアント：サーバーツールの呼び出し

```
asyncio.run(
    call_mcp_tool(
        "convert_timezone",
        {"date_time": "2025-01-20T14:30:00",
         "from_timezone": "America/New_York",
         "to_timezone": "Asia/Tokyo"}
    )
)
```

```
'2025-01-21T04:30:00+09:00'
```

練習しましょう！

MODEL CONTEXT PROTOCOL（MCP）入門